

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
24 April 2003 (24.04.2003)

PCT

(10) International Publication Number
WO 03/034215 A2

- (51) International Patent Classification?: **G06F 9/45**
- (21) International Application Number: **PCT/IB02/04305**
- (22) International Filing Date: 16 October 2002 (16.10.2002)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:
0125176.8 19 October 2001 (19.10.2001) GB
- (71) Applicant: **KONINKLIJKE PHILIPS ELECTRONICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).
- (72) Inventor: **WILLIS, Ian**; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (74) Agent: **WHITE, Andrew, G.**; Internationaal Octrooibureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).
- (81) Designated States (*national*): CN, JP, KR.
- (84) Designated States (*regional*): European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK, TR).
- Published:
— without international search report and to be republished upon receipt of that report.
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

BEST AVAILABLE COPY**WO 03/034215 A2**

(54) Title: A METHOD OF COMPILING BYTECODE TO NATIVE CODE

(57) Abstract: A method of compiling byte code into native code disclosed together with a server (100) and a receiving terminal (101) for the same. The method comprises the steps of transmitting byte code files containing byte code from a server to a receiving terminal; transmitting grouping information relating to the grouping of byte code files to the receiving terminal; and compiling byte code files received at the receiving terminal to native code. The byte code files are selected for compilation as a function of the grouping of byte code files.

DESCRIPTION

A METHOD OF COMPILING BYTECODE TO NATIVE CODE

5 This invention relates to a method of compiling byte code into native code and in particular where the byte code is contained in byte code files transmitted from a server to a receiving terminal for subsequent compilation.

10 Digital Video Broadcasting (DVB) is an organisation which undertakes standardisation activities in the field of digital broadcasting (see <http://www.dvb.org>). DVB's Multimedia Home Platform (MHP) standard (ETSI reference TS 101 812, version 1.1.1, blue book reference A057 05/00) is a series of measures designed to promote the harmonised transition from analogue television (TV) to a digital interactive multimedia future. In particular,
15 the MHP defines a generic, i.e. hardware independent, interface between interactive digital applications and the terminals on which those applications execute. This enables digital content providers to address all types of terminals ranging from low-end to high-end set top boxes, integrated digital TV sets and multimedia PCs. The MHP supports many kinds of applications including the
20 following examples: electronic program guides, information services (enhanced teletext, news and stock tickers etc.), applications synchronised to TV content, gaming and e-commerce.

 The DVB MHP standard and its composite elements are known and well documented, and therefore only those elements which relate to the
25 present invention are elaborated upon hereafter.

 Digital Storage Media Command and Control (DSM-CC) is employed in DVB's MHP as a toolkit for developing control channels associated with MPEG-1 and MPEG-2 streams. Principally defined in part 6 of the MPEG-2 standard (Extensions for DSM-CC), it enables amongst other things broadcast
30 carousels to be defined. Such carousels are mechanisms for assembling data sets into a buffer which are played-out in a cyclic manner, i.e. broadcast periodically. For example, such a carousel may be used to broadcast all the

application files including Java class files, textual information and graphical data required for the receiver to execute MHP applications.

The DSM-CC Object Carousel is used not only in DVB's MHP but also in some digital terrestrial television broadcasts including, for example, ONdigital, BBC TV and Channel 4 digital broadcasts in the UK.

MHP is based around a platform known as DVB-J and includes a virtual machine as defined in the Java Virtual Machine (JVM) specification from Sun Microsystems. Java provides a means of producing hardware independent, executable "byte code" which is executed on a JVM. Different hardware platforms can have different JVM implementations but all will execute the same byte code. This is ideal for broadcast environments such as digital TV, where applications can be written, compiled and broadcast to a variety of different set-top boxes or integrated digital TVs.

Java source code is compiled with a Java compiler into byte code and stored in class files, which can be executed by a JVM. With DVB's MHP, Java application class files are broadcast to a receiver using DSM-CC which may prescribe a corresponding file structure which can then be replicated to store or cache the class files at the receiver. In addition, the DSM-CC may be used to defining modules comprising groups of related Java class files and other files for broadcast.

In the first JVM implementations, the byte code was interpreted however modern JVMs compile the byte code into faster executing, native instructions. Such compilation may typically occur in any of the following ways:

Just in time (JIT)

This is where byte code is compiled to native code either when a class file is loaded or a method (a section of byte code which performs a certain function) within a class file is executed. A disadvantage of JIT implementations is that the JVM must spend an amount of time compiling a class or method when it is first accessed. During this time the JVM waits until the compilation is completed, pausing the execution of the Java application. Typically the more

optimisation that is performed by the compiler the larger the latency, but the faster the native code will run.

Runtime analysis

5 These are based on an interpreted JVM. The JVM takes measurements at runtime and assesses which classes or methods would produce the best performance gain if they were compiled. This could include measuring the number of times a class or method is used and the amount of time spent executing byte code from that class or method. Once identified these can be
10 compiled in the foreground (introducing latency) or compiled in the background, using otherwise wasted processor time. JVMs using this technology have less latency when new class files are loaded or previously unused methods are executed than JIT implementations, but run at a slower speed since some interpretation is always performed. The storage of runtime
15 information has a small overhead on the execution speed. This technique requires both a byte code interpreter and a byte code to native code compiler.

Ahead of time (AOT)

20 The byte code is processed after compilation and a new class file is produced. The new class file contains native implementations of methods. The JVM recognises these native implementations and uses them instead of byte code. Typically better optimisations can be performed by AOT time compilation than JIT since compile time does not hold up the execution of the Java application. The native code is specific to a single processor, thus AOT
25 compilation is not well suited to a broadcast environment since there is likely to be wide range of processors used by different receiver manufacturers.

30 It is an object of the invention to provide an alternative and efficient method of compiling byte code contained in byte code files transmitted from a server to a receiving terminal into native code.

According to the present invention, such a method comprising the steps of transmitting byte code files containing byte code from a server to a receiving terminal, e.g. using a broadcast carousel; transmitting grouping information relating to the grouping of byte code files to the receiving terminal; and
5 compiling byte code files received at the receiving terminal to native code, wherein the byte code files are selected for compilation as a function of the grouping of byte code files. After compilation, the byte code files may be cached in a memory for subsequent execution.

The method of the present invention enables selected byte code to be
10 compiled ahead of time whilst still retaining platform independence, and without the latency associated with JIT compilation or run-time analysis.

The grouping information may relate to the grouping of the byte code files for transmission; the directory structure of the byte code files as stored at the server; or the directory structure to be used to store the byte code files at
15 the receiving terminal.

In the case of the first option, the byte code files may be selected for compilation upon either the interpretation, compilation or execution of other byte code files transmitted in the same group. For example, such grouping information may be described using DSM-CC protocols of the DVB's MHP
20 standard.

In the case of the later two options, a single group would typically relate to a single directory such that byte code files may then be selected for compilation upon the interpretation, compilation or execution of other byte code files residing in the same directory.

25 Again in the case of the later two options, the grouping information may comprise file locations appended to the file name of each respective byte code file, i.e. a new use of information which is conventionally already broadcast.

Further provided in accordance with the present invention is a server as claimed in claims 14 to 18 and receiving terminal as claimed in claims 19 to
30 27.

The present invention will now be described, by way of example only, with reference to the accompanying schematic figures in which:

Figure 1 shows a TV broadcasting system and corresponding receiving system configured in accordance with the present invention;

5 Figures 2 and 3 shows respective directory and module grouping structure of Java class files and other related files as utilised by the broadcasting system of figure 1.

Figure 1 shows a TV broadcasting system 100 adapted to transmit TV
10 audio and video data together with ancillary application data to a corresponding receiving system 101 in accordance with the DVB's MHP standard. For example, the TV broadcasting system may be a satellite broadcasting system belonging to a commercial satellite broadcaster and the receiving system may be one of many corresponding combinations of a
15 satellite receiving dish and a set-top box for processing the received data for display on a TV 116. Equally, the broadcasting system and corresponding receiving system could be terrestrially based. In either case, the design, manufacturing and operation of such systems are well known and those parts which do not relate to the present invention will not be elaborated upon here
20 further.

Through audio 105, video 104 and data 106 inputs, the broadcasting system 100 is fed audio and video data streams from storage media 102 containing pre-recorded audio and video TV content and an application data stream from a computer 103 comprising Java class files and related graphics
25 files. The application data is for providing an enhanced data / teletext service (not shown) which includes ancillary information associated with the audio and video content, and through which information the user can navigate whilst the audio and video data is displayed at the same time.

The audio, video and data streams are multiplexed in a multiplexer 107
30 before being broadcasted using a transmitter 108 over a single channel to corresponding receiving systems. In the receiving system shown, the incoming transmissions are received by a receiver 109, demultiplexed in a demultiplexer

110 and audio, video and application data streams are outputted through respective audio 112, video 111 and data 113 outputs. Display data derived from the application data is then feed with the audio and video data in to a TV signal processing unit 115 from which a corresponding TV signal is outputted
5 for display on a TV 116.

The Java class files of the application data may be provided for, amongst other things, generating windows in which to display the ancillary information associated with the audio and video content, and associated graphics files may be provided for aesthetics including for displaying
10 decorative framing windows and menu navigation options. An example of such an enhanced data / teletext service might be where the audio and video content represents a performing pop group and the application data provides a data / teletext service with a web style interface to be displayed in a window over part of the audio and video display, and in which fans of the pop group
15 may peruse pages of information details forthcoming concerts, gossip and the like.

Certain Java class files will be related to each other to the extent that the interpretation or compilation of one class file will result or is likely in the interpretation or compilation of another. I.e. one class file is likely to call for
20 execution another. This information could be obtained by run-time analysis at the receiver but as previously stated, this introduces latency. Fortunately however, upon authoring the application data, it will be appreciated which class files are so related and these class files can be grouped accordingly. In accordance with the present invention, these groupings are identified by the
25 receiving system whereby certain class-files are pre-compiled and cached in a memory for later execution upon either the interpretation, compilation or execution of other byte code files in the same group.

The groupings of the class files may be conveyed to the receiving system in accordance with either of the following examples:

30

Example 1

The grouping is defined in relation to the arrangement of the byte code files during transmission. Such grouping may be defined using the DSM-CC of DVB's MHP which is able to group class files and other files in to modules for broadcast wherein the contents of a module are associated with a particular application. For example, referring to figure 2, module 1 contains a list of all the class files in directory "dir1"; module 2 contains class files "file1.class", "file4.class", "file5.class" and a list of class files in directory "dir2"; and module 3 contains class files "file2.class", "file3.class", "file6.class", "file7.class" and a list of all class files in directory "dir2", and also graphics file "file8.gif". Byte code files may then be selected for compilation upon either the interpretation, compilation or execution of other byte code files transmitted in the same group.

Example 2

The grouping is defined by the directory structure of the byte code files as stored at the server prior to transmission, and whereby a single group corresponds to a single directory. In this case, the grouping information may be distributed amongst the byte code files in that it may comprise file locations appended to the file name of each respective byte code file. For example, referring to figure 3, the directory structure shown may correspond to 3 groups of class files, group 1 corresponding to directory "dir1" which contains all the class files; group 2 corresponding to "dir2" which contains class files "file4.class" and "file5.class"; and group 3 corresponding to "dir3" which contains class files "file6.class" and "file7.class". In this case, a given class file can belong to more than one group although it need only be broadcast once. Byte code files may then be selected for compilation whereby those class files in the groups corresponding to directories "dir2" and "dir3" are selected upon the interpretation, compilation or execution of other byte code files residing in the same directory; and upon compilation of class files "file1.class", "file2.class" and "file3.class" (i.e. those in directory "dir1" but not directories "dir2" or "dir3"), all the class files in directory "dir1" (including those in "dir2" and "dir3") are compiled.

Example 3

Similar to example 2 except that the grouping is defined by the directory structure to be used to store the byte code files at the receiving terminal which need not be the same as the directory structure of the files as stored at the server.

5

Example 4

The grouping is defined by additional information transmitted with the class files and provided specifically for the sole purpose of defining groups of class files for efficiently compilation.

10

The embodiments above describe the invention primarily in the context of a broadcast system broadcasting byte code on a carousel. However, the invention is equally applicable to other arrangements where byte code is transmitted to a receiving system including Internet transfers of Java applets and the like.

15

CLAIMS

1. A method of compiling byte code into native code comprising the steps of:

- 5 - transmitting byte code files containing byte code from a server to a receiving terminal;
- transmitting grouping information relating to the grouping of byte code files to the receiving terminal; and
- 10 - compiling byte code files received at the receiving terminal to native code,

wherein the byte code files are selected for compilation as a function of the grouping of byte code files.

2. A method according to claim 1 wherein the grouping information
15 relates to the grouping of the byte code files for transmission.

3. A method according to claim 2 wherein the byte code files selected for compilation are done so upon either the interpretation, compilation or execution of other byte code files transmitted in the same group.

20

4. A method according to claim 2 or claim 3 wherein the grouping information is defined using the Digital Storage Media Command and Control (DSM-CC) protocols of the Digital Video Broadcasting's Multimedia Home Platform standard.

25

5. A method according to claim 1 wherein the grouping information relates to the directory structure of the byte code files as stored at the server whereby a single group relates to a single directory.

30

6. A method according to claim 5 wherein the grouping information comprising file locations appended to the file name of each respective byte code file.

7. A method according to claim 5 or claim 6 wherein the byte code files selected for compilation are done so upon the interpretation, compilation or execution of other byte code files residing in the same directory.

5

8. A method according to claim 1 wherein the grouping information defines the directory structure to be used to store the byte code files at the receiving terminal whereby a single group relates to a single directory.

10

9. A method according to claim 8 wherein the grouping information comprising file locations appended to the file name of each respective byte code file.

15

10. A method according to claim 8 or claim 9 wherein the byte code files selected for compilation are done so upon the interpretation, compilation or execution of other byte code files residing in the same directory.

20

11. A method according to any of the preceding claims wherein the server transmits byte code files to the receiving terminal using a broadcast carousel.

25

12. A method according to any of the preceding claims wherein the byte code files selected for compilation are cached in a memory after compilation for subsequent execution.

13. A method according to any preceding claim wherein the byte code is Java byte code.

30

14. A server comprising a transmitter adapted to transmit byte code files containing byte code to a receiving terminal and grouping information relating to the grouping of byte code files wherein the grouping information is transmitted specifically for the purpose of specifying groups of byte code files

whereby the byte code files are selected for compilation into native code as a function of the grouping of byte code files.

15 15. A server according to claim 14 wherein the grouping information relates to the directory structure of the byte code files as stored at the server whereby a single group relates to a single directory.

10 16. A server according to claim 15 wherein the grouping information comprising file locations appended to the file name of each respective byte code file.

17. A server according to any of claims 14 to 16 adapted to transmit the byte code files to the receiving terminal using a broadcast carousel.

15 18. A server according to any of claims 14 to 17 wherein the byte code is Java byte code.

20 19. A receiving terminal comprising a receiver adapted to receive both byte code files containing byte code and grouping information relating to the grouping of byte code files from a server, and a compiler for compiling the byte code files received at the receiving terminal to native code, wherein the byte code files are selected for compilation as a function of the grouping of byte code files.

25 20. A receiving terminal according to claim 19 wherein the grouping information relates to the grouping of the byte code files as received by the receiver.

30 21. A receiving terminal according to claim 20 wherein the byte code files selected for compilation are done so upon either the interpretation, compilation or execution of other byte code files in the same group.

22. A receiving terminal according to claim 20 or claim 21 wherein the grouping information is defined using the Digital Storage Media Command and Control (DSM-CC) protocols of the Digital Video Broadcasting's Multimedia Home Platform standard.

5

23. A receiving terminal according to claim 19 wherein the grouping information defines the directory structure to be used to store the byte code files at the receiving terminal whereby a single group relates to a single directory.

10

24. A receiving terminal according to claim 23 wherein the grouping information comprising file locations appended to the file name of each respective byte code file.

15

25. A receiving terminal according to claim 23 or claim 24 wherein the byte code files selected for compilation are done so upon the interpretation, compilation or execution of other byte code files residing in the same directory.

20

26. A receiving terminal according to any of claims 19 to 25 wherein the byte code files selected for compilation are cached in a memory after compilation for subsequent execution.

27. A receiving terminal according to any of claims 19 to 26 wherein
25 the byte code is Java byte code.

1/2

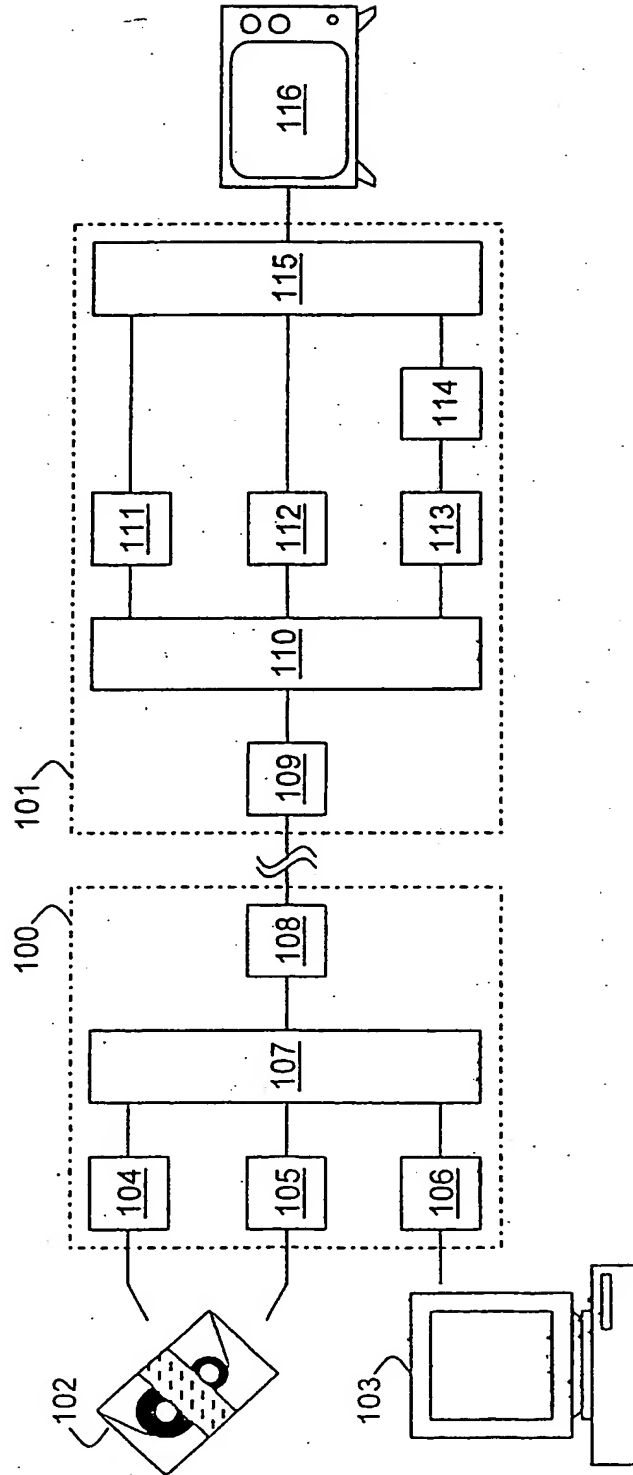
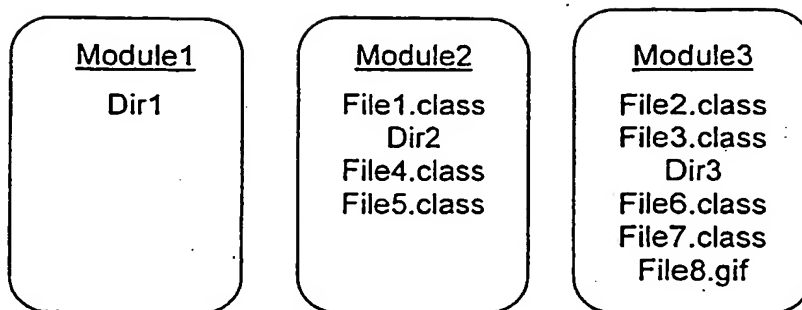
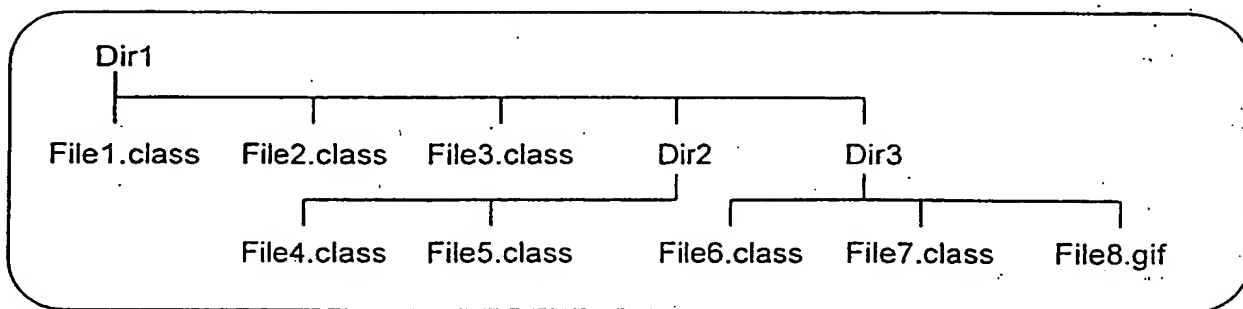


FIG. 1

2/2

FIG. 2FIG. 3

THIS PAGE BLANK (USPTO)

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
24 April 2003 (24.04.2003)

PCT

(10) International Publication Number
WO 2003/034215 A3

- (51) International Patent Classification⁷: **G06F 9/45**
- (74) Agent: **WHITE, Andrew, G.**; Internationaal Octrooi-
bureau B.V., Prof. Holstlaan 6, NL-5656 AA Eindhoven
(NL).
- (21) International Application Number:
PCT/IB2002/004305
- (81) Designated States (*national*): CN, JP, KR.
- (22) International Filing Date: 16 October 2002 (16.10.2002)
- (84) Designated States (*regional*): European patent (AT, BE,
BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, IE, IT,
LU, MC, NL, PT, SE, SK, TR).
- (25) Filing Language: English
- (26) Publication Language: English
- Published:
— with international search report
— before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments
- (30) Priority Data:
0125176.8 19 October 2001 (19.10.2001) GB
- (71) Applicant: **KONINKLIJKE PHILIPS ELECTRON-
ICS N.V.** [NL/NL]; Groenewoudseweg 1, NL-5621 BA
Eindhoven (NL).
- (88) Date of publication of the international search report:
22 January 2004
- (72) Inventor: **WILLIS, Ian**; Prof. Holstlaan 6, NL-5656 AA
Eindhoven (NL).
- For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.*

(54) Title: A METHOD OF COMPILING BYTECODE TO NATIVE CODE

(57) Abstract: A method of compiling byte code into native code disclosed together with a server (100) and a receiving terminal (101) for the same. The method comprises the steps of transmitting byte code files containing byte code from a server to a receiving terminal; transmitting grouping information relating to the grouping of byte code files to the receiving terminal; and compiling byte code files received at the receiving terminal to native code. The byte code files are selected for compilation as a function of the grouping of byte code files.

WO 2003/034215 A3

INTERNATIONAL SEARCH REPORT

Internation Ication No

PCT/IB 02/04305

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/45

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>PLEZBERT: "Continuous Compilation for Software Development and Mobile Computing" 'Online! May 1996 (1996-05) XP002258008 Retrieved from the Internet: <URL: http://citeseer.nj.nec.com/cache/papers/cs/364/http:zSzzSzwww.cs.wustl.edu:zSzcSzSztechreportszSz1996zSzwucs-96-21.pdf/plezbert96continuous.pdf> 'retrieved on 2003-10-14! page 4, line 17 -page 5, line 2 page 9, line 1 - line 14 page 12 -page 18</p> <p style="text-align: center;">-/-</p>	<p>1,2,4-6, 8,9, 11-20, 22-24, 26,27</p>

☒ Further documents are listed in the continuation of box C.☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

T later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

X document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

Y document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

G document member of the same patent family

Date of the actual completion of the international search

10 November 2003

Date of mailing of the international search report

25/11/2003

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Dewyn, T

INTERNATIONAL SEARCH REPORT

International Publication No
PCT/IS 02/04305

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>PLEZBERT M P ET AL: "DOES JUST IN TIME = BETTER LATE THAN NEVER?" CONFERENCE RECORD OF POPL '97: 24TH. ACM SIGPLAN-SIGACT SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGE. PARIS, JAN. 15 - 17, 1997, CONFERENCE RECORD OF POPL: ACM SIGPLAN-SIGACT SYMPOSIUM ON PRINCIPLES OF PROGRAMMING LANGUAGE, NEW YORK, ACM, US, vol. CONF. 28, 15 January 1997 (1997-01-15), pages 120-131, XP000741815 ISBN: 0-89791-853-3 page 1 -page 6</p>	<p>1,2,4-6, 8,9, 11-20, 22-24, 26,27</p>
A	<p>EP 1 076 290 A (SUN MICROSYSTEMS INC) 14 February 2001 (2001-02-14) the whole document</p>	<p>1-27</p>
A	<p>US 6 295 641 B1 (BROWN MICHAEL WAYNE ET AL) 25 September 2001 (2001-09-25) column 4</p>	<p>1-27</p>
A	<p>EP 0 908 819 A (SUN MICROSYSTEMS INC) 14 April 1999 (1999-04-14) column 5, line 41 - line 56 column 9, line 56 -column 10, line 49</p>	<p>1-27</p>
A	<p>FUHRHOP, KRAFT, KUBIS: "Object Carousel Simulator for Broadcast Applications" 'Online! September 1999 (1999-09) XP002258009 Retrieved from the Internet: <URL: http://www.fokus.gmd.de/research/cc/magic/employees/christian.fuhrhop/egmm99_broadcast_simulator_final.pdf> 'retrieved on 2003-10-14! Eurographics Multimedia'99; Milan,Italy, 7-8. September 1999. the whole document</p>	<p>1-27</p>

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/IB 02/04305

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
EP 1076290	A	14-02-2001	US 2003105888 A1 EP 1076290 A2	05-06-2003 14-02-2001
US 6295641	B1	25-09-2001	NONE	
EP 0908819	A	14-04-1999	US 5995754 A CN 1234551 A EP 0908819 A2 JP 11272476 A US 6237141 B1	30-11-1999 10-11-1999 14-04-1999 08-10-1999 22-05-2001

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☒ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☒ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☒ **SKEWED/SLANTED IMAGES**
- ☒ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☒ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

This Page Blank (uspto)